Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ОТЧЕТ

по лабораторной работе №1

Тема работы: Расчет функции

Выполнил

студент: гр. 151003 Матошко И.В

Проверил: Фадеева Е.П.

Минск 2021

Содержание

[1 Постановка задачи 3](#_Toc85493584)

[2 Структура данных 4](#_Toc85493585)

[ПРИЛОЖЕНИЕ А 5](#_Toc85493586)

[ПРИЛОЖЕНИЕ Б 9](#_Toc85493587)

# Постановка задачи

Вычислить количество высших, простых монахов и монахов-учеников при заданных значениях монахов(m𝑜𝑛) и пирогов(pir), если каждому высшему монаху полагается заданное количество 𝑃𝑖𝑟𝑉 пирогов, простому 𝑃𝑖𝑟𝑃𝑟 пирогов, а ученику 𝑃𝑖𝑟𝑈𝑐ℎ пирогов.

Вывести на печать результаты расчётов:

Количество высших монахов, количество простых монахов, количество монахов-учеников, количество получившихся вариантов.

# Структура данных

Таблица 1 -Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| NumV | Integer | Количество высших монахов |
| NumPr | Integer | Количество простых монахов |
| NumUch | Integer | Количество монахов-учеников |
| i | Integer | Количество вариантов |
| mon | Integer | Количество всех монахов |
| Proverka1 | String | Переменная для проверки введенных данных |
| prom | Extended | Переменная промежуточного вычисления |
| pir | Extended | Количество всех пирогов |
| PirV | Extended | Количество пирогов для одного высшего монаха |
| PirPr | Extended | Количество пирогов для одного простого монаха |
| PirUch | Extended | Количество пирогов для одного монаха-ученика |
| proverka | Boolean | Переменная для проверки условий ввода |

ПРИЛОЖЕНИЕ А

(обязательное)

Исходный код программы

Program Project1;

{ Output the number of monks of each type, if the

leading monk eats 10 pies, the simple one has 5 pies,

and the students eat 0.5 pies.The number of all monks and pies

is entered from the keyboard }

{$APPTYPE CONSOLE} // Console application

uses

System.SysUtils;

var

NumV, NumPr, NumUch, mon, i: integer;

Proverka1: string;

prom, pir, pirV, PirPr, PirUch : extended;

proverka:boolean;

{ NumV, NumPr, NumUch - the number of monks of each species;

pir - the number of all pies;

mon - the number of all monks;

i - number of options;

Proverka1 - check for input data;

prom - intermediate calculations;

pirV, PirPr, PirUch - the amount that each kind of monk eats;

proverka - checker for input data. }

Begin

// Check for input all monks

proverka := false;

repeat

writeln('Введите количество монахов');

readln(Proverka1);

if trystrtoint(Proverka1, mon) = true then

begin

mon := strtoint(Proverka1);

if mon < 0 then

writeln(' Значение должно быть положительным')

else

proverka := true;

end

else

writeln('Некорректный ввод данных');

until proverka = true;

// Check for all input pies

proverka := false;

repeat

writeln('Введите количество пирогов');

readln(Proverka1);

if trystrtofloat(Proverka1, pir) = true then

begin

pir := strtofloat(Proverka1);

if pir < 0 then

writeln(' Значение должно быть положительным')

else

proverka := true;

end

else

writeln('Некорректный ввод данных');

until proverka = true;

// A series of checks for the number of pies consumed

// by each type of monk

proverka := false;

repeat

writeln('Введите количество пирогов, которое ест 1 высший монах');

readln(Proverka1);

if trystrtofloat(Proverka1, pirV) = true then

begin

pirV := strtofloat(Proverka1);

if pirV < 0 then

writeln(' Значение должно быть положительным')

else

proverka := true

end

else

writeln('Некорректный ввод данных');

until proverka = true;

proverka := false;

repeat

writeln('Введите количество пирогов, которое ест 1 простой монах');

readln(Proverka1);

if trystrtofloat(Proverka1, PirPr) = true then

begin

PirPr := strtofloat(Proverka1);

if PirPr < 0 then

writeln(' Значение должно быть положительным')

else

proverka := true;

end

else

writeln('Некорректный ввод данных');

until proverka = true;

proverka := false;

repeat

writeln('Введите количество пирогов, которое ест 1 монах-ученик');

readln(Proverka1);

if trystrtofloat(Proverka1, PirUch) = true then

begin

PirUch := strtofloat(Proverka1);

if PirUch < 0 then

writeln(' Значение должно быть положительным')

else

proverka := true;

end

else

writeln('Некорректный ввод данных');

until proverka = true;

// Number of variants

i := 0;

// iterate over the number of high monks

for NumV := 0 to trunc((pir / pirV)) do

begin

// solution of the system of equations

prom := pir - PirUch \* mon - NumV \* (pirV - PirUch);

// the number of simple monks must be whole and greater than 0

if ((frac(prom / (PirPr - PirUch))) < 0.0000001) then

begin

NumPr := trunc(prom / (pirPr - PirUch));

// the number of monks must be whole and greater than zero

if (NumPr >= 0) then

begin

// the number of monks disciples

NumUch := mon - NumPr - NumV;

if (NumUch >= 0) then

begin

// counting the number of options

i := i + 1;

writeln('');

writeln('Высших монахов - ', NumV);

writeln('Простых монахов - ', NumPr);

writeln('Монахов учеников - ', NumUch);

end;

end;

end;

end;

writeln('');

writeln('Всего количество вариантов :', i);

readln;

End.

ПРИЛОЖЕНИЕ Б

(обязательное)

Тестовые наборы

Тест 1

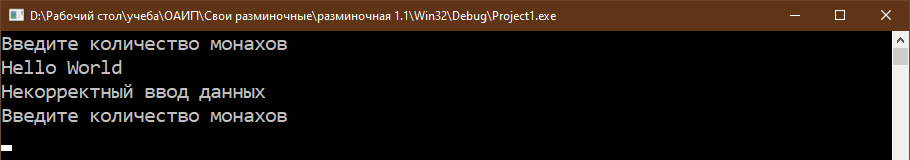
Тестовая ситуация: проверка ввода букв в поле для чисел

Исходные данные: mon = ‘Hello World’

Ожидаемый результат:

Некорректный ввод данных

Полученный результат:



Тест 2

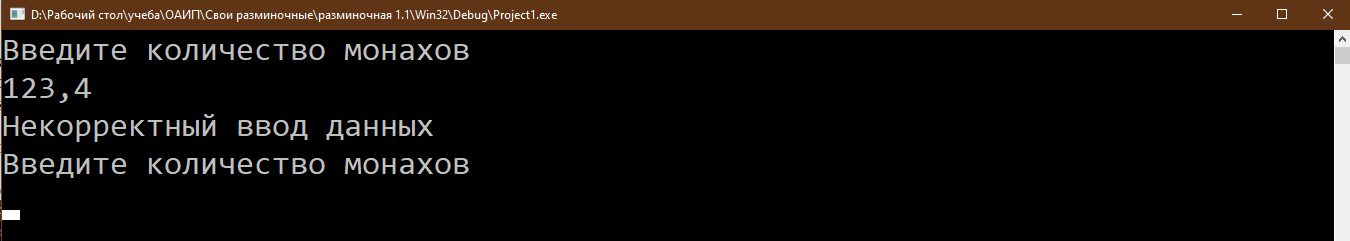
Тестовая ситуация: проверка ввода нецелочисленных значений

Исходные данные: mon = ’123,4’

Ожидаемый результат:

Некорректный ввод данных

Полученный результат:



Тест 3

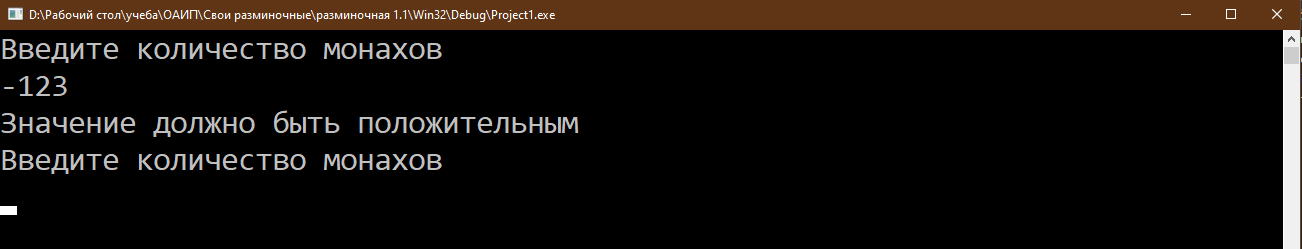
Тестовая ситуация: проверка ввода отрицательных значений

Исходные данные: mon = ’-123’

Ожидаемый результат:

Значение должно быть положительным

Полученный результат:



Тест 4

Тестовая ситуация: проверка решений

Исходные данные: mon = ’100’, pir =’100’, PirV =’10’, PirPr =’5’,

PirUch = ‘0,5’.

Ожидаемый результат:

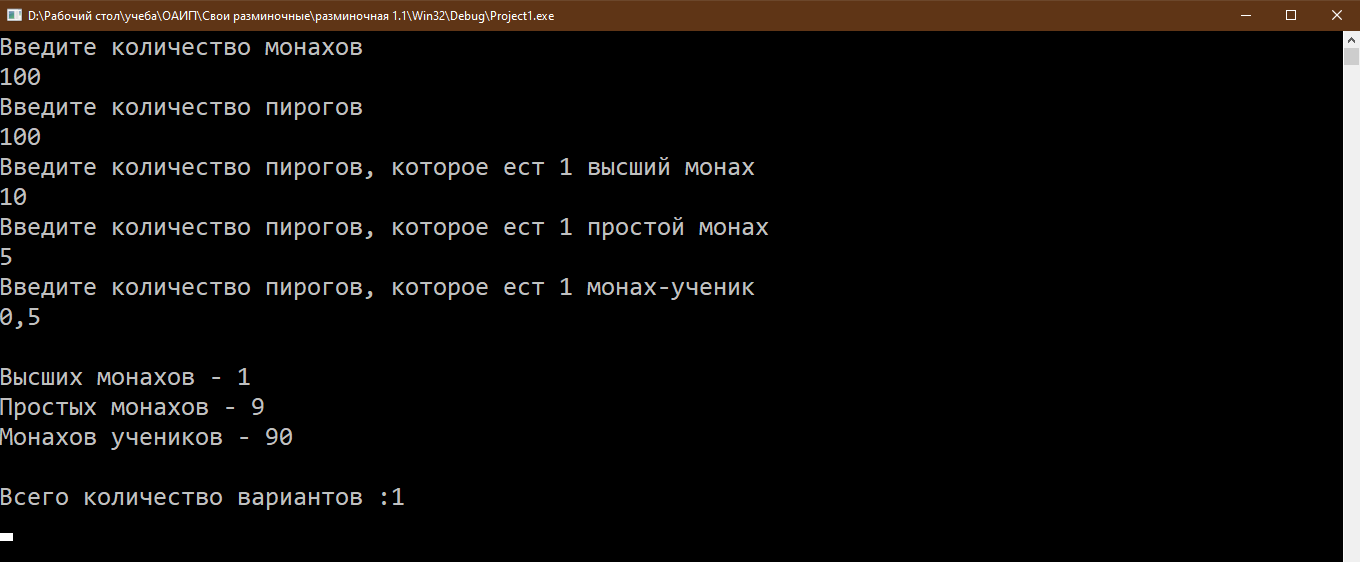
Высших монахов: 1

Простых монахов: 9

Монахов-учеников: 90

Количество вариантов: 1

Полученный результат:



Тест 5

Тестовая ситуация: проверка решений, если ученики едят больше чем высшие

Исходные данные: mon = ’74’, pir =’72,5’, PirV =’0,25’, PirPr =’2’,

PirUch = ‘2,5’.

Ожидаемый результат:

Высших монахов : 44

Простых монахов : 27

Монахов-учеников : 3

Высших монахов : 46

Простых монахов : 18

Монахов-учеников : 10

Высших монахов : 48

Простых монахов : 9

Монахов-учеников : 17

Высших монахов : 50

Простых монахов : 0

Монахов-учеников : 24

количество вариантов : 4

Полученный результат:

